APPLICATION FOR PATENT

Table of Contents

TITLE OF INVENTION

"Mechanism to sift through search results using keywords from the results."

INVENTOR

Abraham Fowler
5757 E. University Blvd.
Apt. 27J
Dallas, TX 75206
(214) 378-9128
abraham@fowler.org
US citizen

CROSS-REFERENCE TO RELATED APPLICATIONS

I filed a provisional patent application for the present invention on May 1, 2001. The number for that provisional application is 60/287,369. It was filed under the same inventor name and invention title as the present invention.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not applicable.

REFERENCE TO A COMPUTER PROGRAM LISTING APPENDIX

This specification includes a computer program listing appendix, submitted with the application in CD-R format. Two copies are submitted with this application. The contents of each CD-R are as follows:

Filename	Size in bytes	Date
Readme.txt	2,037	1/25/02
dist/ThistleSifter.jar	40,664	1/25/02
docs/Application for Patent.doc	84,480	1/25/02
docs/drawings/Backside of drawings.vsd	19,968	1/5/02
docs/drawings/Figure 1.vsd	70,656	1/5/02
docs/drawings/Figure 2.vsd	159,744	1/5/02
docs/drawings/Figure 3.vsd	133,120	1/5/02
src/manifest.txt	45	1/25/02
src/com/thistlesifter/Keyword.java	3,047	1/15/02
src/com/thistlesifter/KeywordList.java	9,784	1/15/02
src/com/thistlesifter/KeywordMetrics.java	1,523	1/14/02
src/com/thistlesifter/KeywordSifter.java	4,485	1/14/02
src/com/thistlesifter/Result.java	4,709	1/15/02
src/com/thistlesifter/ResultList.java	3,800	1/15/02
src/com/thistlesifter/Search.java	19,758	1/15/02
src/com/thistlesifter/Sifter.java	871	1/14/02
src/com/thistlesifter/SiftOperation.java	3,406	1/14/02
src/com/thistlesifter/SimpleSifter.java	1,076	1/14/02
src/com/thistlesifter/ThistleSifter.java	20,657	1/25/02
src/com/thistlesifter/util/TagReader.java	6,271	1/14/02

Transmittal Letter for CD-R Appendix

First Inventor: Abraham Michael Fowler
USPTO Customer Number: 29022
Invention Title: "Mechanism to sift through search results using keywords from the results"

Format of the CD-R: ISO 9660 standard
Format of the files: IBM-PC compatible (Microsoft Windows 95 and up)
Creation date: January 25, 2002

The contents of each of the two CD-R copies are identical to each other. The contents of each CD-R are as follows:

Filename	Size in bytes	Date
Readme.txt	2,037	1/25/02
dist/ThistleSifter.jar	40,664	1/25/02
docs/Application for Patent.doc	84,480	1/25/02
docs/drawings/Backside of drawings.vsd	19,968	1/5/02
docs/drawings/Figure 1.vsd	70,656	1/5/02
docs/drawings/Figure 2.vsd	159,744	1/5/02
docs/drawings/Figure 3.vsd	133,120	1/5/02
src/manifest.txt	45	1/25/02
src/com/thistlesifter/Keyword.java	3,047	1/15/02
src/com/thistlesifter/KeywordList.java	9,784	1/15/02
src/com/thistlesifter/KeywordMetrics.java	1,523	1/14/02
src/com/thistlesifter/KeywordSifter.java	4,485	1/14/02
src/com/thistlesifter/Result.java	4,709	1/15/02
src/com/thistlesifter/ResultList.java	3,800	1/15/02
src/com/thistlesifter/Search.java	19,758	1/15/02
src/com/thistlesifter/Sifter.java	871	1/14/02
src/com/thistlesifter/SiftOperation.java	3,406	1/14/02
src/com/thistlesifter/SimpleSifter.java	1,076	1/14/02
src/com/thistlesifter/ThistleSifter.java	20,657	1/25/02
src/com/thistlesifter/util/TagReader.java	6,271	1/14/02

```
 Volume in drive D:\ is Patent
 Directory of D:\dist\

ThistleSifter.jar                                           40 KB         1/25/02

  1 file(s)
  Total filesize 40 KB
  0 kilobytes free
 Volume in drive D:\ is Patent
 Directory of D:\docs\

Application for Patent.doc                              83 KB         1/25/02

   1 file(s)
   Total filesize 83 KB
   0 kilobytes free
 Volume in drive D:\ is Patent
 Directory of D:\docs\drawings\

Backside of drawings.vsd                                  20 KB        1/5/02
Figure 1.vsd                                              70 KB        1/5/02
Figure 2.vsd                                              157 KB       1/5/02
Figure 3.vsd                                              131 KB       1/5/02

  4 file(s)
  Total filesize 375 KB
  0 kilobytes free
 Volume in drive D:\ is Patent
 Directory of D:\src\

manifest.txt                                      1 KB        1/25/02

  1 file(s)
  Total filesize 1 KB
  0 kilobytes free
 Volume in drive D:\ is Patent
 Directory of D:\src\com\thistlesifter\

Keyword.java                                            3 KB         1/15/02
KeywordList.java                                        10 KB        1/15/02
KeywordMetrics.java                                     2 KB         1/14/02
KeywordSifter.java                                      5 KB         1/14/02
Result.java                                             5 KB         1/15/02
ResultList.java                                         4 KB         1/15/02
Search.java                                             20 KB        1/15/02
Sifter.java                                             1 KB         1/14/02
SiftOperation.java                                      4 KB         1/14/02
SimpleSifter.java                                       2 KB         1/14/02
ThistleSifter.java                                      21 KB        1/25/02

  11 file(s)
  Total filesize 72 KB
  0 kilobytes free
 Volume in drive D:\ is Patent
 Directory of D:\src\com\thistlesifter\util\

TagReader.java                                        7 KB         1/14/02

   1 file(s)
   Total filesize 7 KB
   0 kilobytes free
```

FIELD OF THE INVENTION

This invention is related, in general, to the searching of collections of documents of any kind. More specifically, it relates to a method for the user to sift through the results of a search on a collection of documents without viewing each one, using keywords extracted from or otherwise related to the documents.

BACKGROUND OF THE INVENTION AND PRIOR ART

Searching a large collection of documents for information about which we know only a small part has always been difficult. Although this invention addresses the searching of any kind of collection of documents, the best place to illustrate its necessity is the World Wide Web. Search engines help us find pages, but all too often they deluge us with far too many results. For example, let us say a novice puzzle enthusiast wants to try his hand at cracking some good old-fashioned ciphers. To start finding basic information, he goes to Altavista[1] and searches on the keyword "codes". Instead of getting pages on codes and ciphers, which is what he wants, he gets "about 2,459,295" results on topics such as law, genetics, programming, inventory control, telephone area codes, Bible codes, video game cheats... and the list goes on. Sifting manually through all these results to find exactly what he wants would be quite a daunting task for our puzzle enthusiast.

Part of this general problem is that our searches are rarely specific enough, usually because we do not know the proper terminology for our topic that we could use to narrow the search down without inadvertently excluding some relevant results. Human language adds to the problem because it is often ambiguous— for any given search keyword, there can be a whole spectrum of meanings. Only an expert in the desired topic might know the appropriate keywords

[1] Altavista - http://www.altavista.com/

needed to narrow down the search, and even an expert may have trouble finding the results he wants among all the irrelevant results. Let us return to our example. If our puzzle enthusiast had known more about his topic, he might have searched instead on the keywords "cipher" or "cryptography". The keyword "cipher" would have narrowed down the results to 17,364 pages (at Altavista); combining it with the original keyword "codes" would narrow it down even further to 3,488 pages. In the enthusiast's case, choosing the correct keywords to narrow down the search required a greater depth of knowledge about the topic than he already had. The irony is that if he had deeper knowledge on the topic, he might not be searching the Web in the first place.

Prior inventions have attempted to aid the user by grouping the results of a broad initial search into subcategories. Patent no. 6,167,397, "Method of clustering electronic documents in response to a search query," describes one such method, which involves having the computer look through each document and algorithmically discover similarities between groups of documents in the search results. Many World Wide Web search engines now appear to have some incarnation of this method. Although this method is an improvement in that it helps the user avoid some documents that are obviously unrelated, it suffers from the same basic problem of the ambiguity of human language and the mere fact that no algorithm really understands the meaning of the document it is processing. As a result, these machine-generated categories often appear artificial and at times even misleading.

Patent no. 5,924,090, "Method and apparatus for searching a database of records," describes an attempt to improve further on this by using human-generated categories that have been established in a database beforehand. It attempts to fit each of these documents into one of these human-generated categories, again by a sophisticated algorithm on the document itself. This approach can be seen implemented at Northern Light's search website.[2] The approach is a

[2] Northern Light – http://www.northernlight.com/

little better than having machine-generated categories, but still falls short in that it is limited to only the pre-established categories and the knowledge of those who created them. The World Wide Web is far too vast for any human effort to fully categorize, and new categories are popping up all the time. Moreover, this approach again suffers from the same problem of the ambiguity of human speech and sometimes places documents in the wrong categories.

Another invention, patent no. 6,012,053, "Computer system with user-controlled relevance ranking of search results," attempts to help users bring the documents they seek to the top of the list of search results, by giving them a number of "relevancy factors" which they can control to give each individual document a "relevancy score" or ranking within the list. These relevancy factors could be things such as size of the file, date of creation, location of search terms, proximity of search terms to each other, and so on. However, by focusing only on the search terms in the query, these relevancy factors seem not to take the documents' other contents into consideration, which could be the best indicator of relevancy to the user.

Thus, while all these inventions are an improvement upon the basic search, there is still plenty of room for alternative and possibly better solutions. At present, no known mechanism allows the user to sift the results of a search based on all the keywords extracted from the results themselves, including keywords not found in the original search query. This invention pertains to such a sifting mechanism, one that allows user-controlled reordering and excluding of search results, based on keywords found in the results themselves.

BRIEF SUMMARY OF THE INVENTION

The present invention attempts to give the user a better way to sift through the large volume of search results returned by a broad or general search query. In simplified steps, it does so by: 1) conducting a search using an internal or external search engine back-end, or otherwise receiving search results from some search query; 2) extracting keywords from each search result, for example by reading predefined keywords associated with each result by its author; 3) putting these keywords in a separate list and presenting it to the user alongside the list of search results themselves; 4) allowing the user to choose a *sifting operation* to apply to each keyword, for example to "include" results that match a particular keyword or to "exclude" results that match an undesirable keyword; 5) sifting and reordering the list of search results according to the user's choice of keywords and sifting operations; and 6) optionally resubmitting a more targeted, more refined version of the original search query by adding to the original query the user's choice of keywords to include or exclude.

Some of the steps briefly outlined above can be conducted in several different ways. For example, any search engine that produces adequate results, or even multiple search engines, may be used as the back end. Similarly, any algorithm that extracts useful keywords from the result documents can be used in the second step. In step three, formulating the list of keywords or search results, one particular implementation could perform simple or advanced statistical analysis to determine the optimal presentation order. Another implementation might contain algorithms that, before presenting keywords to the user, could remove keywords deemed insignificant or even perform grammatical analysis to group keywords which are different forms of the same word. The sifting operations presented to the user for keywords could encompass several different levels of "inclusion" or "exclusion," such as 1) "require" this keyword (Boolean AND), 2) "include" documents containing this keyword (Boolean OR), and 3) "exclude" all documents containing this keyword (Boolean AND NOT). Other sifting operations, such as aggregating results or keywords, could also be applied.

Regardless of all these and other potential enhancements, the overall mechanism for sifting results based on keywords and sifting operations remains the same. It is the overall sifting mechanism that constitutes the present invention.

BRIEF DESCRIPTIONS OF THE SEVERAL VIEWS OF THE DRAWING

Figure 1 is a diagram of the information flow between entities at four stages (a, b, c, d) of the process; in this diagram, each shape represents an entity in the system, arrows represent the flow of information, and the text next to the arrows describes the information flow.

Figure 2 is a high-level flowchart giving a general view of the steps in the process. Each step in the flowchart is labeled with a letter.

Figure 3 illustrates a prototypical user interface for the present invention; where (a) shows the overall interface including the list of keywords and search results, and (b) is a cutaway showing the sifting operations available for each keyword.

DETAILED DESCRIPTION OF THE INVENTION

Before delving into the full description of the invention, it will be helpful to define certain terms precisely, as follows:

Search Engine: any software module or program which can search a collection of documents using a query and return a list of matching results.

Search Result: a "hit" or single match for a particular query from a collection of documents.

Keyword: any word or phrase which could be used in a search for a document, and which pertains closely to the subject of the document. Ideally associated directly with the document itself by the author or categorizer of the document, though it could come from an algorithm analyzing the document itself.

Sifting: The process of going through the search results, excluding unwanted results based on keywords and user-selected *sifting operations* on those keywords, and arranging the remaining search results in order of number of included keywords or some other ranking algorithm.

Sifting Operation: Any operation associated with a keyword, which causes a document to be included in, excluded from, or ranked within the list of search results, based on the keyword and its presence or absence in that document.

Having defined the above terms, let us move on to the full description of the invention. The description of the functioning of the present invention is split into four main phases, for convenience and clarity. (The division into four phases here is merely didactical; it is not meant to be understood as intrinsic or necessary to the design or function of the invention.)

In the first phase, the present invention begins with a search query, input by the user or otherwise passed to the invention. The broader and more general the query, the more useful the present invention will ultimately be. At this point, the invention could potentially reformulate the search query to suit a particular search engine, especially if the present invention was programmed to use more than one search engine, or perform some other optimization on the query. The search query is then passed to the search engine back-end, which does its own internal processing and comes up with a list of search results. The flow of information in phase one is illustrated in Figure 1, (a), and the general programmatic steps taken by the present invention are laid forth in Figure 2, (a) (b) and (c).

In the second phase, the search engine back-end has processed the search query and returned a list of search results to the present invention. The present invention then processes this list of search results to gather keywords from each result, using any of a variety of algorithms. The keywords gathered are then compiled into a single list of keywords. After processing, the present invention presents the list of keywords and the list of search results to the end user. The flow of information in phase two is illustrated in Figure 1, (b), and the general programmatic steps taken by the present invention are laid forth in Figure 2, (d) (e) and (f). Let us examine the flowchart steps from Figure 2 a little more closely.

In Figure 2, (d), the present invention uses any one of various algorithms to extract keywords from the search results. The most ideal way to get good keywords is for each document to have a list of associated keywords. Library book cataloguing systems associate keywords with each book, for example, and World Wide Web pages coded in HTML (Hyper Text Markup Language) can contain *meta-tags* which list the keywords for the page. In such cases, the extraction algorithm is simply to add the keywords from each document to the present invention's master list of keywords, optionally compiling other statistics, such as frequency of keyword occurrence, in the process.

In Figure 2, (f), the present invention presents the list of keywords and the list of search results to the user. This can be done through any suitable user interface, such as a Java form, an HTML page, a wireless phone display, a library computer terminal, and so on. Figure 3, (a) illustrates the interface used by the prototype of the present invention to display these two lists to the user. The scope of this invention is not limited by the choice of user interface, as long as the interface can suitably present these two lists to the user, and interactively accept input from the user, as described in more detail in phase three below.

In the third phase, the user interacts with the present invention by selecting a keyword from the keyword list, and choosing from several sifting operations available for that keyword. The most basic sifting operations are "include" and "exclude." During the sifting of documents, these keywords and sifting operations would determine a particular result's inclusion in or exclusion from the list of search results. The present invention then uses the keywords and operations selected by the user to re-sift the list of search results. The sifting process will exclude certain results based on the selected keywords and sifting operations, and may optionally reorder the remaining search results based on other keywords and sifting operations. After sifting the list of search results, the present invention then re-displays them to the user. The user can then choose to interact further with the list of keywords, or move on to view a search result or refine the search. This flow of information is illustrated in Figure 1, (c), and the general programmatic steps taken by the present invention are laid forth in Figure 2, (g) (h) (i) (j) and (k). Let us examine the steps from Figure 2 a little more closely.

In Figure 2, (g), many different user interfaces could be employed to allow the user to interact with the keyword and search result lists. For example, in Figure 3, (b) the present invention's initial prototype displays next to each keyword a "combo box" or dropdown list containing the various sifting operations. A command-line interface, to use a very different example, could use typed text commands to select keywords to operate on in sifting the list of search results. As long as the user interface allows the user to select a keyword and its sifting operation, that user interface satisfies the requirements of the step in Figure 2, (g). The example user interfaces given here are illustrative only, and should be understood not to limit the scope of this invention.

Also in Figure 2, (g), the user must choose a keyword and a sifting operation to use with that keyword. The sifting operation will be used in Figure 2, (h) as part of the sifting algorithm, so the set of operations used in (h) should be made accessible to the user in (g). Typically, the minimal set of available sifting operations are to "include" or "exclude" from the list of search results those documents containing the given keyword. However, any operation which can be used with the keyword to select, rank, or exclude a document within the sifting algorithm can be defined. For example, the present invention's initial prototype— see Figure 3, (b)— provides four sifting operations, named "require," "include," "ignore," and "exclude." In the case of the prototype, "require" represented a Boolean AND between the selected keyword and each document's list of keywords; "include" represented a Boolean OR; "ignore" represented to ignore the keyword when re-sifting; and "exclude" represented a Boolean AND NOT between the selected keyword and each document's list of keywords.

In Figure 2, (h), the present invention takes the list of user-selected keywords and the sifting operation associated with each one, and uses it to run a sifting algorithm on the original list of search results to produce a new, derived list of sifted search results. The effect of the sifting algorithm is to exclude unwanted results from the derived list, and optionally to reorder the ones left according to some ranking given by the sifting operations and keywords, or given by the sifting algorithm itself, or given by some combination of the two. In the prototype, those documents with the greatest number of "required" and "included" keywords were put at the top of the re-sifted list of search results.

In Figure 2, (i), the user is presented with the sifted (derived) list of search results, in a manner similar to that described for Figure 2, (f) before. If the user has chosen their keywords and sifting operations well, they should see a much more relevant set of documents, with the most relevant to their search right at the top of the list.

In Figure 2, (j), the user has a choice to go back to step (g) and choose another keyword and sifting operation to further refine the sifting of the search results, or to continue on to perform other operations with the search. If the user chooses to go back, this step creates an interactive cycle by which the user can continually experiment with including, excluding, etc. different keywords until they achieve the sifted results they like. Note that with a graphical user interface, the choice in (j) should not necessarily be shown explicitly to the user as a separate step; instead, merely providing various buttons or user interface elements, each with its own function, will allow the user to navigate this step without explicitly being asked to choose.

Before continuing on to phase four, it should be clarified that each cycle through phase three may be cumulative; that is, the present invention may, if designed to do so, remember the keywords and sifting operations selected in previous cycles. A graphical user interface lends itself particularly well to displaying the sifting operation associated with each keyword at any given time, and makes this "cumulative" effect intuitive to the user— see Figure 3, (a) and (b). With such an interface, in order to return to the original unsifted list of search results, the user must remove sifting operations from keywords (or set them to a sifting state such as "ignore," in the prototype), or the invention could provide a "clear all sifting operations" function to do this for the user.

The user may decide to stop at phase three by selecting one of the search results to view. In such a case, the present invention displays the document or calls the appropriate functions to cause the document to be displayed. Then it may either exit, or return to the interactive cycle of phase three. Figure 2, (m) and (n) lay forth the general programmatic steps the present invention takes for this task, although those steps do not show how a graphical user interface may exit from or return to the interactive cycle of phase three.

The user may also choose to exit phase three by issuing the "refine search" command to resubmit a new query based on their keyword selections. This command takes them directly to phase four and does not allow them to re-enter phase three until they pass through phases one and two again. A discussion of phase four follows.

The invention enters the fourth phase after the user gives the "refine search" command. In this phase, the invention combines the keywords and sifting operations they chose into the original search query, to produce a new search query more targeted to the topic they desire. For example, if the original search query was for the term "bond", and the user selected the keyword "007" and applied the "exclude" sifting operation to it, the reformulated query string for one particular search engine could be "bond -007" (where the minus symbol tells the search engine to exclude documents with the keyword "007"). This function is especially useful with extremely large collections of documents such as the World Wide Web, since the first search query is likely to be limited by a maximum document count threshold and thus not return all possible or useful matches the first time around. When the search query is formulated, it is passed back to the step in Figure 2, (b) from phase one, and the entire cycle of the present invention begins again. This flow of information is illustrated in Figure 1, (d), and the general programmatic steps taken by the present invention are laid forth in Figure 2, (k) and (l). The fourth phase, which is optional, is the final phase of the overall framework of this invention.

To give a brief example of all this at work, let us return to our puzzle enthusiast. He could start out with a broad query on the word "codes". The search engine would probably return only the first few hundred results out of the millions it cites. The present invention would extract keywords from each search result, and present the search results alongside a separate, cumulative list of the keywords extracted from the results. Our user could then choose to include or exclude selected keywords, which would result in the search results being sifted and redisplayed based on those keywords. In this way the user could intelligently narrow his search without excluding relevant results. In our enthusiast's specific example, he might find it helpful to see that some pages share the keyword "law", others share the keyword "gene", still others the keyword "open source", and so on. He could "exclude" all such keywords, and immediately see search results with those keywords disappear. He might see the keywords "cipher" or "cryptography" among the list and realize those would be good keywords to narrow his search, selecting them and using the "include" sifting operation. Immediately, search results with those keywords would come to the top of the list. Not only would he get better, more specific results, but in the process he would learn something about the proper terminology for his desired topic. Thus, one of the most helpful aspects of this invention is that it shows the user what other keywords may be available relating to their topic.

Now, following is a description of several alternatives, variations or potential improvements to parts of the present invention. Each paragraph below discusses one such improvement or variation, relating it back to the description of the overall mechanism above.

In the first phase, the search engine back-end is mentioned with little discussion of what that may actually be. In the prototype, the user entered a query directly into the present invention, which then submitted the search query to the Altavista[3] World Wide Web search engine. It then processed the HTML document containing the search results, parsing it to extract document title, document location, etc. From this information it loaded the individual World Wide Web pages and extracted the keywords from their meta-tags. This, however, should not be considered to limit the methods this invention could use for receiving search results. For example, it could be augmented to use several World Wide Web search engines, reformulating the search query as necessary for each one, retrieving results from each one, and combining the

[3] Altavista - http://www.altavista.com/

results into one list for processing to extract keywords. Or, the present invention could be tied to a proprietary search engine back-end for searching a private collection of documents, such as in a library book-cataloguing system. Finally, the present invention could also work as an add-on to a search engine, in which case the search engine itself would perform the entire first phase and simply pass the results to the present invention for processing. The examples given here are illustrative only, and should be understood not to limit the scope of this invention, nor broaden it to encompass algorithms or inventions that already stand on their own.

In the flowchart of Figure 2, step (d), the most basic method of extracting keywords from the search results is to take keywords directly associated with the documents, as described before. However, the present invention could also employ other methods for extracting keywords from documents. There exist algorithms that can cull the most important words from a given document, and these algorithms could be plugged into the present invention's architecture. Other algorithms could rely on a saved database of previous user query keywords, associated with the documents they most often chose for those keywords. (For the purpose of sifting later on in phase three, in all cases where keywords are extracted from the document using such algorithms, the present invention should remember the keywords that belong to each document in addition to storing them all in a master keyword list.) In running through the list of search results to gather keywords, the present invention could also modify the list of search results itself. For example, if the implementation is programmed to use only keywords associated with the documents, it could throw out results which do not have any keywords associated with them. To sum up, any means of obtaining keywords from the documents can potentially function inside of the present invention. The examples given here are illustrative only, and should be understood not to limit the scope of this invention, nor broaden it to encompass algorithms or inventions that already stand on their own.

Also in the flowchart of Figure 2, step (d), during the extraction of keywords, the present invention could gather statistical information about the keywords and documents, for use in the sifting operations or the sifting algorithm later. For example, the invention could count the number of documents with which each keyword is associated. It could count the number of documents in which two or more keywords appear together. It could gather information on which keywords appeared to be associated most strongly with their documents (by means of repetition within the document, or location of occurrence within the document, for example). In short, any statistics or other information that could be useful to the sifting in later stages may be gathered at this point by the present invention. The examples given here are illustrative only, and should be understood not to limit the scope of this invention, nor broaden it to encompass algorithms or inventions that already stand on their own.

In Figure 2, (e), the present invention can optionally employ algorithms to improve the quality of the keywords. In many cases, a word that represents a single idea can have many forms (plural vs. singular; adjective and verbal forms; etc.). Without any optimization of the list of keywords, these variant forms can clutter the list and make it harder for the user to identify common themes in the keyword list. An example of such an algorithm could be grammatical analysis of words to reduce various forms into one keyword. Another algorithm could combine close synonyms when their meanings were found unambiguous, according to some database or computerized thesaurus. Statistical analysis could be performed on the keyword list, clustering related keywords, bringing common themes to the top, or weeding out keywords unlikely to be chosen, for example. These algorithms can be as simple or as complex as desired. The examples given here are illustrative only, and should be understood not to limit the scope of this invention, nor broaden it to encompass algorithms or inventions that already stand on their own.

In the description of Figure 2, steps (g) and (h), sifting operations such as "include" or "exclude" were mentioned as examples. Other sifting operations could be devised that may prove useful as well. For example, sifting operations which use fuzzy logic, or word counts, or relevance algorithms, or require that the keyword be a central theme in the document, etc. to include, exclude or rank documents could all be used within the framework of the present invention. All that is required is that the operation relate the keyword to the document in some way useful to the sifting algorithm of Figure 2, (h). The examples given here are illustrative only, and should be understood not to limit the scope of this invention, nor broaden it to encompass operations, algorithms or inventions that already stand on their own.

Figure 2, step (h) represents the sifting of the search results using keywords and sifting operations. The sifting may also optionally reorder the results according to some calculated rank given each one, as described before. The ranking algorithm may be cumulative; that is, it may combine the rankings of several different keywords and sifting operations for a single document to produce that document's final ranking in the list. The algorithm may be defined primarily by the sifting operations; it may have some cumulative functions such as counting the number of matched keywords; or it may even involve a much more complicated process within the sifting algorithm such as relating documents to each other, grouping them by topic and/or keyword density, etc. The examples given here are illustrative only, and should be understood not to limit the scope of this invention, nor broaden it to encompass operations, algorithms or inventions that already stand on their own.

It should now be clear to any skilled programmer or software engineer how to put together a system implementing the architecture of the present invention. Many possible different ways of implementing certain parts of the present invention have been set forth, and to repeat, these should not be construed to limit the scope of the invention nor broaden it to encompass pre-existing or independently developed mechanisms, algorithms or inventions. Moreover, the exclusion of a particular algorithm from the list of examples for each of those parts should not be construed as limiting the present invention from using such algorithm. The appended claims which define the scope of this invention are made independent of any and all such complementary algorithms, mechanisms or inventions.